In the Specification:

Change paragraph 0002 as follows :

A technique of time-varying SNR dependent coding for increased communication channel robustness is described by A. Bernard ,one of the inventors herein, and A. Alwan in "Joint channel decoding –Viterbi Recognition for Wireless Applications ", in Proceedings of Eurospeech, Sebt. 2001, vol. 4, pp. 2703-6; A. Bernard, X. Liu, R. Wesel and A. Alwan in "Speech Transmission Using Rate-Compatable Trellis codes and Embedded Source Coding," IEEE Transactions on Communications, vol. 50, no. 2, pp 309-320, Feb. 2002.; A. Bernard and A. Alwan , "Source and Channel Coding for low bit rate distributed speech recognition systems", IEEE Transactions on Speech and Audio Processing, Vol. 10, No. 8, pp570-580, Nov. 2202; and A. ~~Bernand~~ Bernard in "Source and Channel Coding for Speech and Remote Speech Recognition," Ph.D. thesis, University of California, Los Angeles, ~~2001~~ 2002.

Change paragraph 0011 as follows:

In general, there are two related approaches to solve the temporal alignment problem with HMM speech recognition. The first is the application of dynamic programming or Viterbi decoding, and the second id the more general forward/backward algorithm. The Viterbi algorithm (essentially the same algorithm as the forward probability calculation except that the summation is replaced by a maximum operation) is typically used for segmentation and recognition and the forward/backward for training. See for the Viterbi algorithm G.D. ~~Fornay~~ Forney, " The Viterbi algorithm, " IEEE Transactions on Communications, vol. 61, no. 3, pp. 268-278, April 1973.

Change paragraph 0012 as follows:

The Viterbi algorithm finds the state sequence Q that maximizes the probability P^* observing the features sequence ($O = o_1, \ldots$ ~~o_t~~ o_T) given the acoustic model λ

$$P^* = \max_{\text{All } Q} P\,(Q, O \mid \lambda). \qquad (1)$$

Change paragraph 0014 as follows :
The maximum likelihood $P^*(O \mid \lambda)$ is then given by $P^*\ (O \mid \lambda) = \max_j \{\varphi_j(T)\}$.

Change paragraph 0019 as follows:

Under the hypothesis of a diagonal covariance matrix Σ , the overall probability $b_j(o_t)$ can be computed as the product of the probabilities of observing each individual feature. The weighted recursive formula (equation 3) can include individual weighting factors ~~$\gamma_{t,t}$~~ $\gamma_{k,t}$ for each of the N_F front-end features.

$$\varphi_{j,t} = \max\,[\varphi_{i,t-1}\, a_{ij}] \prod_{k=1}^{N_F} [b_j\,(o_t)]^{\gamma_{k,t}}.. \qquad (4)$$

where k indicates the ~~dimension~~ index of the feature observed.

Change paragraph 0022 as follows:
In order to perform time and frequency SNR dependent weighting, we need to change the way the probability $b_j\,(o_t)$ is computed. Normally, the probability of observing the N_F -dimensional feature vector o_t in the j^{th} state is computed as follows,

$$b_j\,(o_t) = \sum_{m=1}^{N_M} w_m \frac{1}{\sqrt{(2\pi)^{N_F}|\Sigma|}} e^{-\frac{1}{2}(o_t-\mu)'\Sigma^{-1}(o_t-\mu)}, \qquad (5)$$

where N_M is the number of mixture components, w_m is the mixture weight, and the parameters of the multivariate Gaussian mixture are its mean vector μ and covariance matrix Σ.

Change paragraph 0023 as follows:

In order to simplify notation, we should ~~only~~ note that $\log(b_j(o_t))$ is proportional to a weighted sum of the cepstral distance between the observed feature and the cepstral mean $(o_t-\mu)$, where the weighting coefficients are based on the inverse covariance matrix (Σ^{-1}),

$$\log(b_j(o_t)) \quad \infty (ot-\mu)' \; \Sigma^{-1} (o_t-\mu). \qquad (6)$$

Change paragraph 0024 as follows:

Remember that the N_F -dimensional cepstral feature o_t is obtained by performing the Discrete Cosine Transform (DCT) on the N_S- dimensional log Mel spectrum (*S*). Mathematically, if the ~~N_S x N_F~~ N_F x N_S dimensional matrix *M* represents the DCT transformation matrix, then we have $o_t = MS$. Reciprocally, we have $S= M^{-1} o_t$ where M^{-1} (N_S x N_F) represent the matrix for the inverse DCT ~~matrix~~ operation.

Change paragraph 0027 to correct the symbol on either side of Σ as follows:

With this notation, the weighted probability of observing the feature becomes

$$\tilde{b}_j(o_t) = \sum_{m=1}^{N_M} w_m \frac{1}{\sqrt{(2\pi)^{N_F}|[\Sigma]|}} e^{-\frac{1}{2}(o_t-\mu)'(MG_tM^{-1})'\Sigma^{-1}(MG_tM^{-1})(o_t-\mu)} \qquad (9)$$

which can be rewritten using a back-and-forth weighted time-varying transformation matrix $T_t= MG_tM^{-1}$ as

$$\tilde{b}_j(o_t) = \sum_{m=1}^{N_M} w_m \frac{1}{\sqrt{(2\pi)^{N_F}|\{\Sigma\}|}} \;\mu) \quad e^{-\frac{1}{2}(o_t-\mu)'(T_t')\Sigma^{-1}(T_t)(o_t-\mu)} , \qquad (10)$$

which can also resemble the unweighted equation 5 with a new inverse covariance matrix

$\tilde{\Sigma}^{-1} = T_t' \, \Sigma^{-1} \, T_{t,}$

$$\tilde{b}_j(o_t) = \sum_{m=1}^{N_M} w_m \frac{1}{\sqrt{(2\pi)^{N_F}\underline{|}\cancel{[}\Sigma\cancel{]}\underline{|}}} e^{-\frac{1}{2}(o_t-\mu)'\tilde{\Sigma}^{-1}{}_t(o_t-\mu)} \tag{11}$$

Change paragraph 0030 as follows:

In that specific case, the time and frequency SNR evaluation we are using for the purpose of evaluating the presented technique is that of the ETSI Distributed Speech Recognition standard ~~[6]~~ which evaluates the SNR in the time and frequency domain for spectral subtraction purposes. See ETSI STQ-Aurora DSR Working Group, "Extended Advanced Front-End (xafe) Algorithm Description," Tech. Rep., ETSI, March 2003.

Change paragraph 0032 as follows:

One particular instantiation of equation 12 is using a Wiener filter type equation applied on the linear SNR estimate to obtain,

$$\gamma_{t,f} = \frac{\sqrt{\eta_{t,f}}}{1+\sqrt{\eta_{t,f}}} \qquad , \underline{\gamma_{tlf} \geq 0}$$

$$\underline{\gamma_{tlf} = 0} \qquad\qquad \underline{\gamma_{tlf} \leq 0}$$

which guarantees that $\gamma_{t,f}$ is equal to 0 when $\eta_{t,f}=0$ and $\gamma_{t,f}$ approaches 1 when $\eta_{t,f}$ is large.

Change paragraph 0033 as follows:

Figure 2 illustrates the block diagram for the time and frequency weighted Viterbi recognition algorithm. When you have speech (speech frame t) the first step 21 is to estimate the SNR to get $\eta_{t,f}$. Then the weighting is calculated to get $\gamma_{t,f}$ at step 23. Then the transform matrix computation at step 25 is performed. This is the MG_tM^{-1} to get ~~Tt~~ $\underline{T_t}$. The next step is Viterbi decoding at step 27 to get $b_j(o_t)$. Here the original MFCC feature o_t is sent to the recognizer. The original feature contains the information about the SNR.